

**AMENDMENTS TO THE CLAIMS**

1. (Currently Amended) A method in a computer system for performing an inter-thread long jump, the method comprising:  
receiving an indication of a set jump location, the set jump location set by a set jump thread;  
determining whether a the set jump thread that set the set jump location is the same thread that is currently executing; and  
when the set jump thread is not the same thread that is currently executing, setting a state of the set jump thread to execute a long jump indicating the set jump location.
2. (Original) The method of claim 1 wherein the setting of the state includes setting a program counter of the set jump thread to point to a long jump routine.
3. (Original) The method of claim 1 wherein when the set jump thread is blocked on an operating system call, aborting the operating system call.
4. (Currently Amended) The method of claim 3 including, after setting the state of the set jump thread, starting the execution of the set jump thread.
5. (Original) The method of claim 1 wherein when the set jump thread is running, notifying the set jump thread of the inter-thread long jump.
6. (Original) The method of claim 1 wherein when the set jump thread is running, directing the set jump thread to enter a known state prior to setting the state of the set jump thread.

7. (Original) The method of claim 6 wherein the known state is a quiescent state.

8. (Original) The method of claim 1 including determining whether the set jump thread still exists.

9. (Original) The method of claim 1 wherein the computer system supports multiple streams.

10. (Currently Amended) A system for performing an inter-thread long jump, the system comprising:

a processing component;

a component executing on the processing component that determines whether a set jump thread that set a set jump location is the same thread that is currently executing; and

a component executing on the processing component that, when the set jump thread is not the same thread that is currently executing, sets a state of the set jump thread to transfer control to the set jump location.

11. (Original) The system of claim 10 wherein the setting of the state includes setting a program counter of the set jump thread to point to a long jump routine.

12. (Original) The system of claim 10 wherein when the set jump thread is blocked on an operating system call, aborting the operating system call.

13. (Currently Amended) The system of claim 12 including a component that, after the state of the set jump thread is set, starts execution of the set jump thread.

14. (Original) The system of claim 10 wherein when the set jump thread is running, notifying the set jump thread of the inter-thread long jump.

15. (Original) The system of claim 10 wherein when the set jump thread is running, directing the set jump thread to enter a known state prior to setting the state of the set jump thread.

16. (Original) The system of claim 15 wherein the known state is a quiescent state.

17. (Original) The system of claim 10 including determining whether the set jump thread still exists.

18. (Original) The system of claim 10 wherein multiple streams are supported.

19. (Original) A method in a computer system for performing an inter-thread long jump, the method comprising:

under control of a set jump function, storing a current stream state, the current stream state including a return address; and

under control of a long jump function, when the long jump function is invoked by a thread that is different from a thread that invoked the set jump function, locating state information for the thread that invoked the set jump function in the restored stream state;

setting a program counter in the located state information to point to an instruction that invokes the long jump function; and

under control of the thread that invoked the set jump function, performing a long jump by executing the instruction pointed to by the program counter wherein the long jump function

deallocates memory allocated since the set jump function was invoked;  
restores the stored stream state; and  
jumps to the return address included in the restored stream state.

20. (Original) The method of claim 19 wherein the current stream state is stored in a set jump buffer.

21. (Original) The method of claim 20 wherein the restoring includes retrieving the stored stream state from the set jump buffer.

22. (Currently Amended) A method in a long jump thread of a computer system for performing an inter-thread long jump to a location in a set jump thread, the method comprising:

in a second thread different from a set jump thread,

receiving information relating to the set jump thread;  
retrieving a thread identifier for the set jump thread from the received information;  
retrieving a thread control block for the set jump thread based on the retrieved thread identifier;  
determining from the retrieved thread control block a state of the set jump thread; and  
causing, based on the determined state, the set jump thread to execute at a set jump location.

23. (Original) The method of claim 22 wherein the state of the thread is blocked, resumable, running, or transition.

24. (Original) The method of claim 23 wherein the thread state is blocked when the thread is blocked on a synchronization timeout.

25. (Original) The method of claim 23 wherein the thread state is resumable when the thread is ready and waiting to be allocated to a stream.

26. (Original) The method of claim 23 wherein the thread state is running when the thread is executing and is not blocked.

27. (Original) The method of claim 23 wherein the thread state is transition when the thread is being allocated a stream.

28. (Currently Amended) The method of claim 23 wherein ~~the modifying includes causing the set jump thread to execute at a set jump location comprises~~ setting a program counter in the thread control block to an address of a long jump routine.

29. (Currently Amended) The method of claim 28 ~~including wherein causing the set jump thread to execute at a set jump location comprises~~ putting the thread control block on a list of unblocked threads.

30. (Currently Amended) The method of claim 28 including in the set jump thread, invoking the long jump routine when the set jump thread starts running.

31. (Currently Amended) The method of claim 23 further comprising:  
when the set jump thread state is blocked,  
removing the set jump thread from a blocked list;  
setting the set jump thread state to resumable; and  
when the set jump thread is blocked on an operating system call,  
aborting the operating system call.

32. (Currently Amended) The method of claim 23 further comprising:  
when the set jump thread state is running,  
when the set jump thread is blocked on an operating system call, aborting the  
operating system call;  
when the thread is handling a data blocked exception, saving additional state  
information of the set jump thread;  
creating a save area data structure;  
initializing the created save area data structure; and  
setting a program counter in the initialized data structure to an address of a  
long jump routine, wherein the long jump routine returns the point of  
execution to the set jump location.

33. (Currently Amended) A system for causing a long jump thread to cause a set  
jump thread to start executing at a location specified by the set jump thread, the system  
comprising:

a processing component;  
means executing on the processing component for specifying a set jump location of  
the set jump thread; and  
means executing on the processing component for, under control of the long jump  
thread, causing the set jump thread to jump to the set jump location.